

## Grundlagen Semantic Web

Seminar für Computerlinguistik, Universität Heidelberg

Sebastian Rudolph

Wintersemester 2009/10

<http://semantic-web-grundlagen.de>

### Übung 4: SPARQL

---

**Aufgabe 4.1** Betrachten Sie das folgende RDF-Dokument mit Informationen über einige Himmelskörper:

```
@prefix ex: <http://example.org/> .
ex:Sonne ex:radius "1.392e6"^^xsd:double ;
        ex:satellit ex:Merkur, ex:Venus, ex:Erde, ex:Mars .
ex:Merkur ex:radius "2439.7"^^xsd:double .
ex:Venus ex:radius "6051.9"^^xsd:double .
ex:Erde ex:radius "6372.8"^^xsd:double ;
        ex:satellit ex:Mond .
ex:Mars ex:radius "3402.5"^^xsd:double ;
        ex:satellit ex:Phobos, ex:Deimos .
ex:Mond ex:name "Mond@de", "Moon@en" ;
        ex:radius "1737.1"^^xsd:double .
ex:Phobos ex:name "Phobos" .
ex:Deimos ex:name "Deimos" .
```

Geben Sie SPARQL-Anfragen an, welche die folgenden Ergebnisse in tabellarischer Darstellung liefern:

1. Objekte, die um die Sonne oder um einen Satelliten der Sonne kreisen.
2. Objekte mit einem Volumen von über  $2 \times 10^{10} \text{ (km}^3\text{)}$  (Himmelskörper mit Radius können als kugelförmig angenommen werden, d.h.  $V = \frac{4}{3}\pi r^3$ ) und, falls vorhanden, dem Objekt, dessen Satellit sie sind.
3. Objekte mit einem Satelliten, für den ein englischsprachiger Name gegeben ist, die außerdem Satellit eines Objektes von über 3000 (km) Durchmesser sind
4. Objekte mit zwei oder mehr Satelliten (nehmen Sie an, dass unterschiedliche URIs hier unterschiedliche Objekte bezeichnen)

**Aufgabe 4.2** Übersetzen Sie die Anfragen aus Aufgabe 4.1 in Ausdrücke der SPARQL-Algebra. Dabei können Join-Operationen mit dem leeren Graphen  $Z$  sofort vereinfacht werden.

**Aufgabe 4.3** Berechnen Sie die Ergebnisse der Ausdrücke aus Aufgabe 4.2 bezüglich der Wissensbasis aus Aufgabe 4.1. Berechnen Sie Teilergebnisse schrittweise wie auf den Folien 24–32 der Vorlesung “Semantic von SPARQL” dargestellt. Stimmen Ihre Ergebnisse mit den erwarteten Resultaten überein?

**Aufgabe 4.4** Durch Kombination von Filtern mit optionalen Graph-Mustern kann man in SPARQL auch nach Elementen suchen, für die eine bestimmte Information *nicht* angegeben ist. Formulieren Sie eine Anfrage nach allen Himmelskörpern, die keinen Satelliten haben.

Gehen Sie dabei davon aus, dass die o.a. Wissensbasis durch Tripel ergänzt wurde, die sämtlichen Himmelskörpern via `rdf:type` den Typ Himmelskörper zuweisen.

**Aufgabe 4.5** Beim Spiel *Sudoku* geht es darum, unvollständige Zahlengitter entsprechend bestimmter Regeln auszufüllen. Wir betrachten folgendes einfaches  $4 \times 4$ -Sudoku:

			3
			4
2			
3			

Ziel ist es, Zahlen von 1 bis 4 so in die Felder einzutragen, dass keine Zahl in einer Zeile, Spalte oder einem der vier  $2 \times 2$ -Teilfelder mehrfach vorkommt. Wir wollen nun SPARQL verwenden, um dieses Problem zu lösen, d.h., jede mögliche Lösung eines Sudokus soll einer durch SPARQL ermittelten Variablenbelegung entsprechen. Versuchen Sie dazu ein geeignetes RDF-Dokument und eine entsprechende SPARQL-Anfrage zu finden. Sollte dies zu anspruchsvoll erscheinen, dann (und nur dann) betrachten Sie noch die nachfolgenden Hinweise.

Da Variablen in SPARQL nur Werte annehmen können, die in einem RDF-Dokument vorkommen, benötigen wir zunächst ein Dokument für die Darstellung der möglichen Zahlenwerte, wie z.B. das folgende:

```
<http://example.org/Feld>    <http://example.org/erlaubt>
    "1"^^xsd:int, "2"^^xsd:int, "3"^^xsd:int, "4"^^xsd:int .
```

Weisen Sie nun jedem der 16 Felder einen Variablennamen zu und formulieren Sie dann die folgenden Bedingungen in Ihrer Anfrage:

- Jedes der Felder nimmt einen Wert an, der eine erlaubte Zahl ist.

- Die vorausgefüllten Felder haben den vorgegebenen Wert.
- Keine zwei unterschiedlichen Felder derselben Zeile sind gleich.
- Keine zwei unterschiedlichen Felder derselben Spalte sind gleich.
- Keine zwei unterschiedlichen Felder desselben Teilquadrats sind gleich.

Die letzten drei Punkte, wörtlich ausgeführt, führen zu wiederholten Bedingungen, die man weiter vereinfachen kann. Da es insgesamt nur wenige mögliche Werte gibt, kann man die Gleichheit und Ungleichheit zweier Werte auch mit Hilfe von weiteren Prädikaten direkt im RDF-Dokument beschreiben und so die Verwendung von Filtern gänzlich vermeiden. Auf ähnliche Weise kann man mit SPARQL beliebige Sudokus lösen (woraus man leicht ableiten kann, dass SPARQL mindestens so schwer wie Sudoku – also NP-hart – ist).

**Aufgabe 4.6** Diese Aufgabe befasst sich mit Modifikatoren in SPARQL. Betrachten Sie folgendes RDF-Dokument:

```
@prefix ex: <http://example.org/> .
ex:a    ex:value  "1"^^xsd:integer ;
        ex:value  "3"^^xsd:integer .
ex:b    ex:value  "2"^^xsd:integer .
```

Welches Ergebnis würde für diese RDF-Datenbasis auf die folgenden SPARQL-Anfragen zurück geliefert?

1. SELECT ?s ?v WHERE { ?s <http://example.org/value> ?v } ORDER BY ?v
2. SELECT ?s WHERE { ?s <http://example.org/value> ?v } ORDER BY ?v
3. SELECT ?s WHERE { ?s <http://example.org/value> ?v } ORDER BY  
DESC(?v) LIMIT 2
4. SELECT DISTINCT ?s WHERE { ?s <http://example.org/value> ?v } ORDER  
BY ?v

Welches Ergebnis würden Sie erwarten, falls der letzten Anfrage LIMIT 1 hinzugefügt würde?